Basic Computer Organization/Architecture

Dr. Edward Nava
ejnava@unm.edu

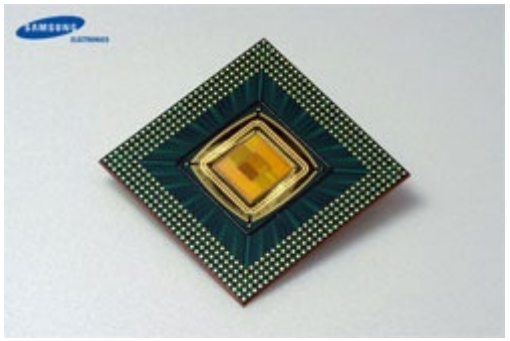
What is a Computer?

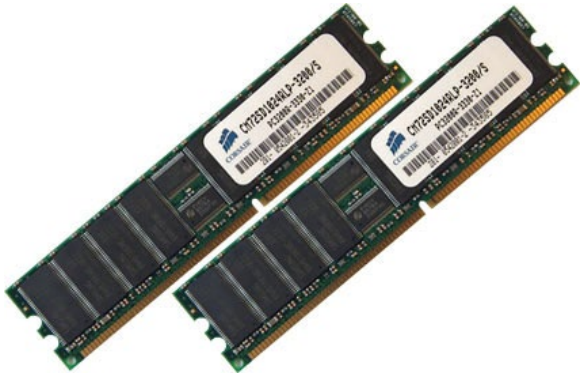
What is a Computer?

- **Programs are stored in the memory unit and instructions are passed to the control unit.**
- **The control unit directs the computer via control lines.**
- **Inputs are passed to the data path unit (registers) which then sends outputs.**
- **The registers are temporary storage units.**
- **An Arithmetic Logic Unit (ALU) performs operations:**
  - Arithmetic
  - Logic
  - Shift
  - Etc.

Major Components of a Computer

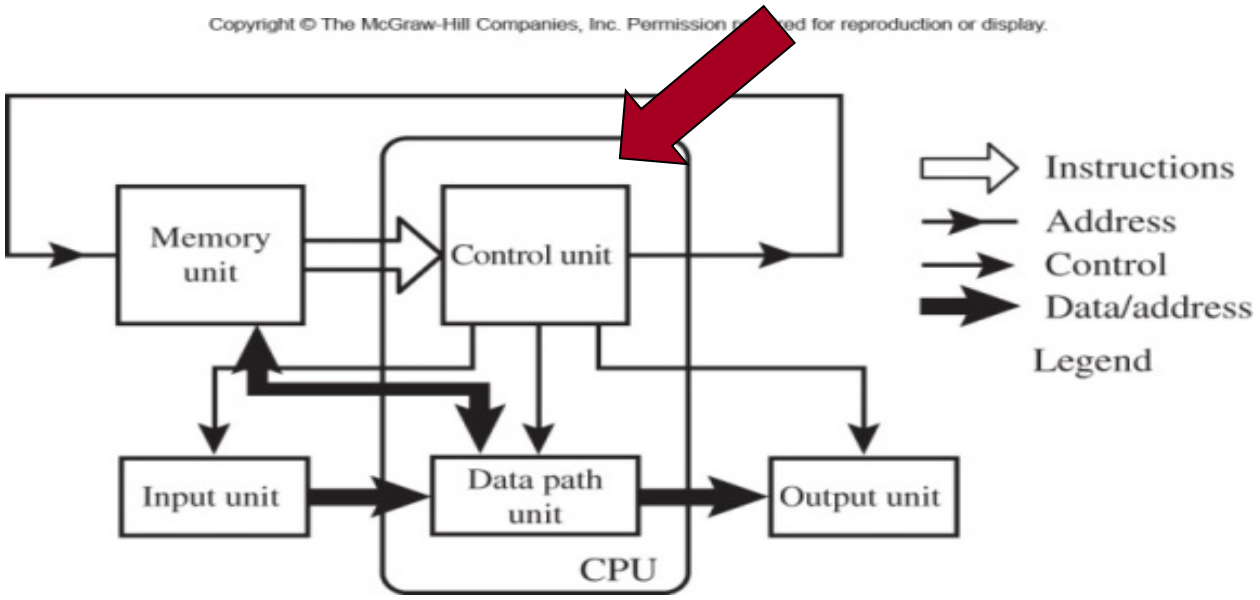
- **Central Processing Unit (CPU)**
- **Random Access Memory (RAM)**
- **Hard Drive / Disk**





CPU (Central Processing Unit)

The CPU is usually the Control Unit and the Data Path Unit.



When the computer you are working with does not have an instruction you need, you can either create an algorithm from existing instructions or add a peripheral that performs the task you need.

What is Computer Architecture?

Definition of Architecture

ar·chi·tec·ture

/ˈärkəˌtek(t)SHər/

noun

1. the art or practice of designing and constructing buildings.
   synonyms: building design, building style, planning, building, construction;
   formal architectonics
   "modern architecture"

2. the complex or carefully designed structure of something.
   "the chemical architecture of the human brain"

Translations, word origin, and more definitions

What is computer architecture?

- **What does "architecture" mean?**
- **Layout and interactions of a computer system**
- **What is a computer system?**
  - Input → Process → Output
- **Can a computer system be more than one computer? Think of an example...**

Types of Computer Architecture

- **Harvard-Type**
  - Howard Aiken
  - Harvard Mark I relay-based computer
  - **Separate Instruction & Data Memory**

- **von Neumann (Princeton)-Type**
  - John von Neumann
  - IAS
  - **Shared memory**

- **Both invented in the 1940s**
- **Today there are many hybrids of both types of architecture**

CISC architecture

- **Fewer registers than RISC machines.**
- **Register-memory architecture: Operands either in registers or in memory.**
- **CISC (Complex Instruction Set Computers) usually have many simple and complex instructions. Instruction length is not fixed. Generally requires more clock cycles to execute.**
- **Many addressing modes available.**

RISC architecture

- **Reduced Instruction Set Computer**
  - Memory accessed by load and store instructions
  - Most operations are Register-to-Register
  - Small number of simple instructions, generally executed in a single machine cycle
  - Usually have a large number of registers
- **Uses few addressing modes (Immediate, Direct, Indirect, etc.)**
  Addressing Mode: Specifies how the computer calculates addresses to locate operands. Different ways in which operands are specified.

Basic Computer Organization/Architecture

Basic Computer Architecture

Cycles: Fetch

Decode

Execute

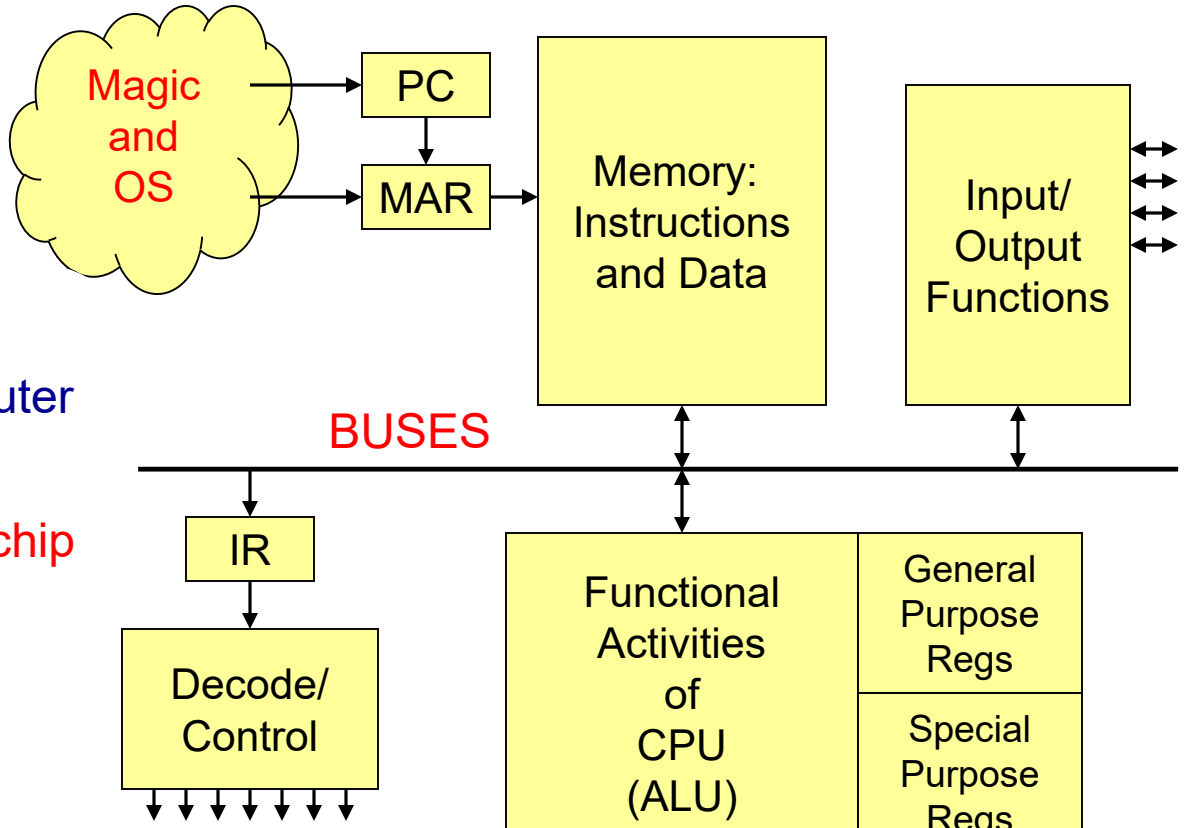
Inside the Computer

BUSES (inside the chip and outside):

Address Bus

Data Bus

Control Bus



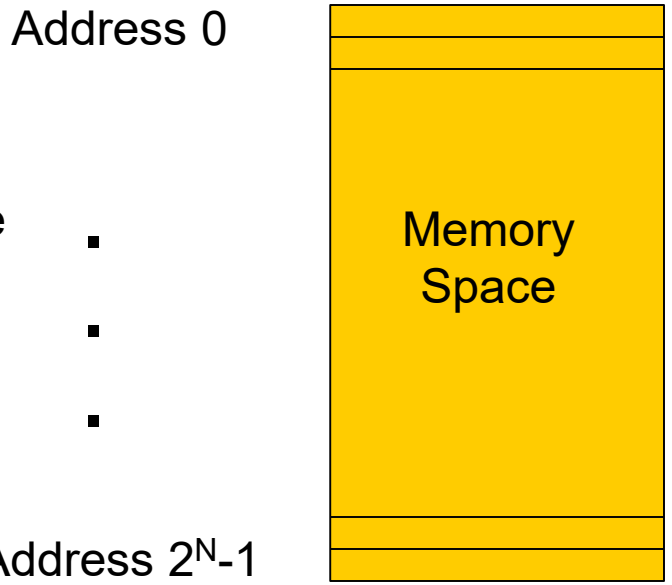
Hardware View: von Neumann (Princeton) -Type

Digital System

Addresses

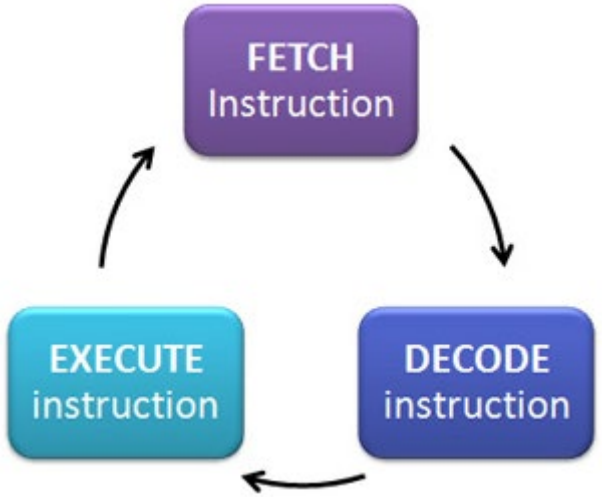
- **Most microprocessors/microcontrollers view memory and I/O devices similarly**
  - **Designer allocates specific addresses to components (Memory – RAM, ROM, I/O devices)**
    - **Memory Map, Memory Map Interfacing, Memory Map I/O, Memory Space Allocation**

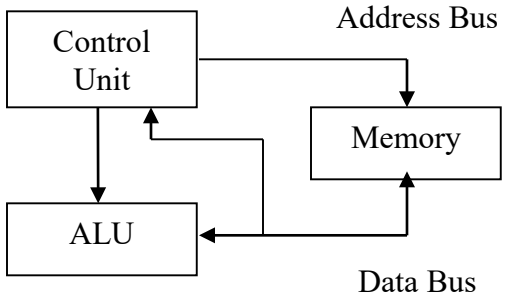
If Address Bus has N bits then we can calculate the Memory Space 2^N = 0 to (2^N-1)



Basic Premise of all Computers

1. **Fetch instruction from memory**
2. **Decode instruction in control unit**
3. **Execute instruction (data may be fetched from memory)**
4. **Store results if necessary**
5. **Repeat!**



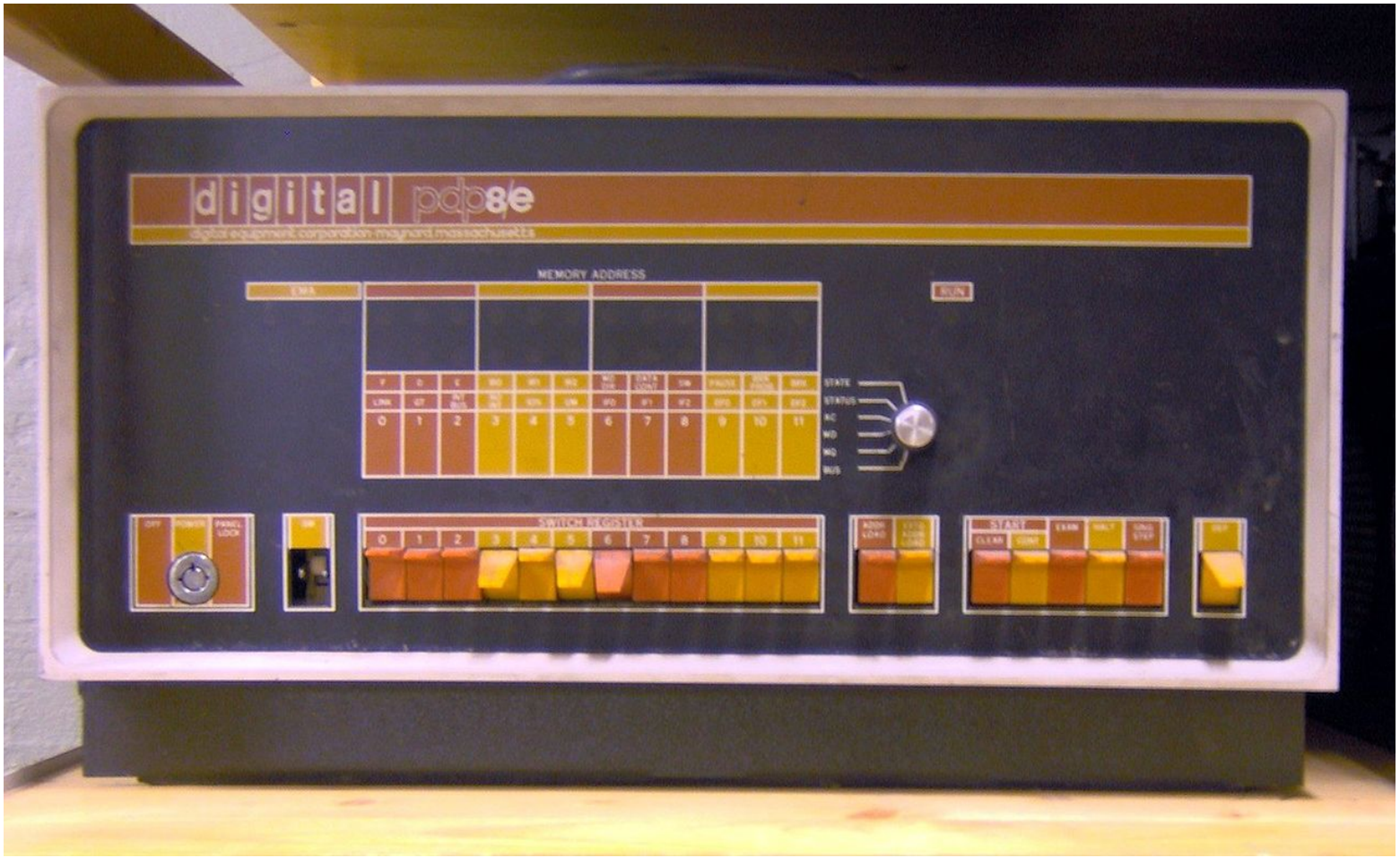


Computer

Programming (Magic)

- **Computers are dumb, they do what they are told to do**
  - **The basic operations of a computer system form what is known as the instruction set**
  - **To solve a problem, you must express the solution to the problem in terms of the instructions of the particular computer**
  - **The approach or method that is used to solve a problem is known as the algorithm**
  - **With an algorithm, you then translate it into a program**
- **Programming**
  - **At the beginning, programming was done in terms of binary numbers (octal, hexadecimal) that correspond directly to the machine instructions and locations in the computer's memory**

PDP-8e


digital pdp8e
MEMORY ADDRESS
RUN
STATE
STATUS
SWITCH REGISTER
START
CLEAR
CONT
EXAM
HALT
SING STEP

PDP11/40

Computer

Programming (Magic)

- **Programming Language Evolution**
  - **Next technological software advance was the development of Assembly languages**
  - **Assembly language (low level language) permits the user to use symbolic names to perform various operations and to refer to the to specific memory locations**
  - **Assembler, a special program that translates the assembly language program from its symbolic format into the specific machine instructions**
  - **One-to-one correspondence between each assembly language statement and a specific machine instruction**
  - **Each microprocessor/microcontroller has its own assembly language; its own instruction set**
  - **Assembly language programs are not portable**

Computer

Programming (Magic)

- **High-Level Languages**
  - **FORTRAN, C, C++, PASCAL, PHP, Python, Ruby, others no longer have to be concerned with the architecture of the particular computer**
  - **Operations performed by these languages are more sophisticated and far removed from the particular instruction set of the particular computer**
  - **One high-level language statement results in many different machine instructions being executed (object code)**
  - **Compiler, translates statements in a high-level language into a form that the computer can understand, that is, into the instruction set of a particular computer (object code)**
  - **Linker, combines the object code produced by the compiler from your C, C++, Java, programs with the object code of routines (EX. I/O) used by your program (libraries)**

Computer

Programming (Magic)

- **High-Level Languages**
  - **Each language is characterized by its syntax and semantics.**
    - **Syntax – comprises of the grammatical rules used to write a program**
    - **Semantics – is the "meaning" associated with language constructs**
  - **When a new computer language is developed, the designers first study the characteristics of the underlying processes and then develop syntactic constructs and their associated semantics to model and express these characteristics.**
  - **Sequential vs. Concurrent programming**
    - **Sequential – one operation at a time**
    - **Concurrent – operations run concurrently**

Computer Programming (Magic)

There are two primary ways for a program to be executed:

Compiler - A program that transforms source code into the machine readable code that a CPU can execute.

Ex: Programs written in C, C++, Java, Fortran and Pascal are typically executed using a compiler.

Interpreter - A program that executes source code directly, line-by-line, as each line of source code is encountered.

Ex: Programs written in Matlab, Ruby, Perl and Python are typically executed using an interpreter.

Computer

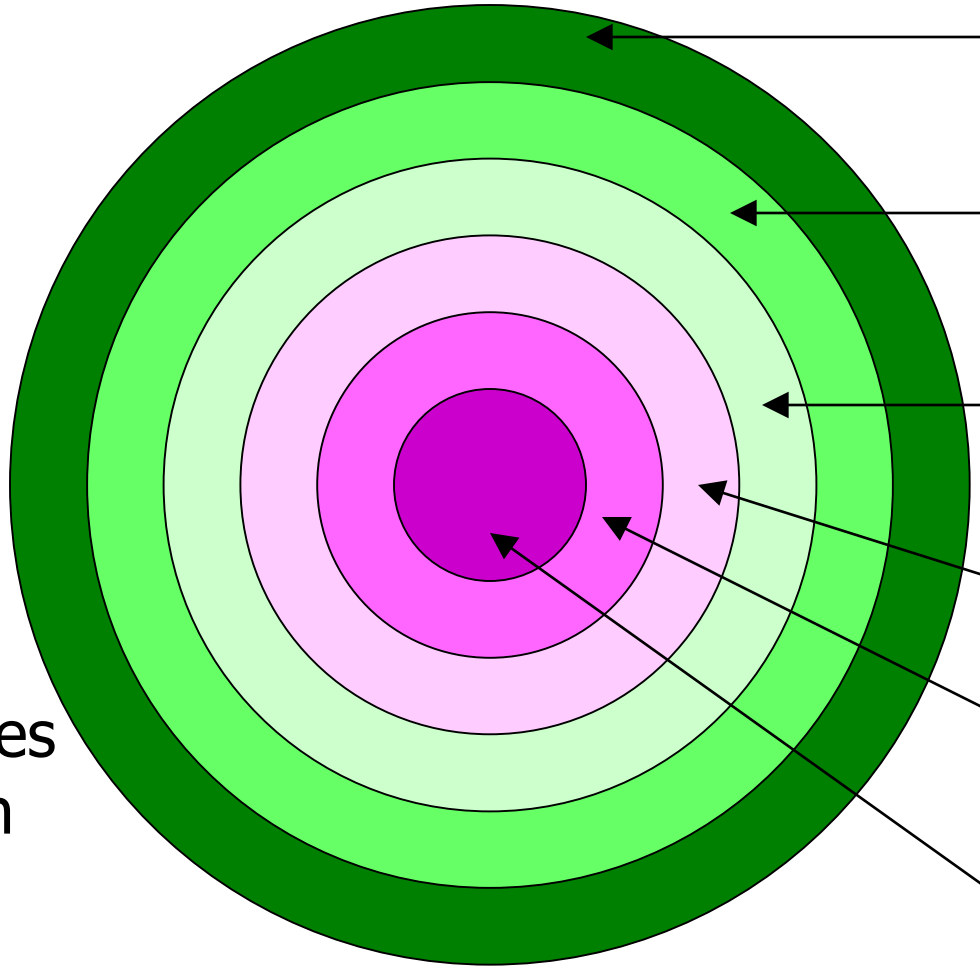
Operating System

- **Is a program that controls the entire operation of a computer system – Manager.**
  - **Manages input and output (I/O) operations**
  - **Manages all the resources available**
  - **Manages the execution of programs – YOURS!!!!**
  - **EX. UNIX, Windows, MAC OS, Android, others**

WEB Services

- **The Internet now has become an innovative software platform.**

Software/Hardware Organization: Another View



Layers

Application

Run-time system
(compiler code, malloc, debugger code, thread management)

Operating system

Register-level h/w

Gate-level h/w

Circuit-level h/w

Interfaces between layers

Types of Languages

- **Programming languages**
  - (e.g., C, C++, assembly language)
- **Scripting languages**
  - Shell scripts (OS) – controls other software programs
  - Emacs, QuakeC
- **Specification languages**
  - Used for system design/analysis
  - UML
- **Machine Code**
  - (the non human-readable form other languages are translated into, sometimes on the fly)
- **Query language**
  - SQL, OQL, XQuery
- **Markup languages**
  - typically used for producing documents
  - HTML, XML, XHTML
- **Transformation languages**
  - transform some input text in a certain formal language into a modified output text that meets some specific goal
  - XQuery
- **Template processing languages**
  - combine one or more templates with a data model to produce one or more result documents
  - Pearl, Python
- **Visual programming languages**
  - Labview, Matlab
- **Hardware description languages**
  - Executables for hardware
  - Verilog, SystemC, VHDL
- **Configuration file formats (e.g., INI file)**

Register Transfer Language (RTL)

RTL for ADD D0, D1

Register Transfer Language

***fetch*:** **MAR ← PC**

PC ← PC+N

IR ← M[MAR]

***decode*:**

***execute*: D1 ← D1 + D0**

Assembly Language Statement

ADD/SUB/MUL/DIV/AND/OR/EXOR